Práctico 7 - Diseño de TDAs

El siguiente práctico se resuelve en papel, ya que es de diseño de implementación.

1. Diseñar el TDA para Punto Cartesiano, donde punto está dado por dos valores x,y £ R tales que el par ordenado (x,y) se puede ubicar en el plano formado por los ejes cartesianos.

#Ejercicio 1

class Puntocartesiano:

def \_\_init\_\_(self, x=float, y=float):

self.x = x

self.y = y

def \_\_float\_\_(self):

imprimir = self.x , self.y

return imprimir

puntocartesiano1 = Puntocartesiano (5,5)

print (puntocartesiano1)

1. Diseñar el TDA para Recta, donde se recuerda que la forma de una recta está dada por y = ax + b donde a es la pendiente y b es la ordenada al origen.

#Ejercicio 2

class Recta:

def \_\_init\_\_(self, a=float, b=float):

self.a = a

self.b = b

x=0

y=0

y = a\*x+b

print (y)

1. Diseñar el TDA para Numero Complejo. ¿Qué operaciones se deben poder realizar so­bre uno, o dos números complejos? Diseñe el diagrama del TDA, sin generar ninguna implementación por el momento.

#Ejercicio 3

class Complejo:

def \_\_init\_\_(self, partereal, parteimaginaria):

self.r = partereal

self.i = parteimaginaria

x = Complejo(3.0, -4.5)

x.r, x.i

(3.0, -4.5)

1. Diseñe el TDA Lógico. (equivalente a los Boolean de los lenguajes en general) Considere las operaciones lógicas que se pueden realizar entre dos valores lógicos, como and, or, xor y la operación unitaria not

#Ejercicio 4

class Lógico:

def \_\_init\_\_(self, partereal, parteimaginaria):

self.not = a

self.and = a\*b

self.or = a+b

self.nand = (ab)

self.nor = a\*b

self.xor = ab+ab

1. Diseñar el TDA Racional. Recordar que un numero racional se puede escribir de la forma p donde p se denomina numerador y q se denomina denominador. Se debe cumplir q = 0

#Ejercicio 5

class Racional:

def \_\_init\_\_():

self.p = numerador

self.q = denominador

x = Racional(p/q)

1. Diseñar el TDA Fecha. Determine que datos contiene y todas las operaciones que debe poder realizar sobre una fecha.

#Ejercicio 6

class Fecha:

def \_\_init\_\_():

self.año = año

self.mes = mes

self.dia = día

x = fecha

1. Si estuvieses diseñando el TDA Lista, ¿qué operaciones le definirías? La implementación de Listas en Python, que operaciones tiene? Googlee y compare el diseño propio con las listas de Python.

#Ejercicio 7

class Lista:

def \_\_init\_\_():

self.suma = suma

self.resta = resta

self.producto = producto

self.cociente = cociente

self.promedio = promedio

x = Lista()

1. Diseñar un TDA Rectángulo, que constará de 3 atributos: largo, ancho y color. Defina las operaciones que un rectángulo debe tener.

class Rectángulo:

def \_\_init\_\_(self, b, h, perímetro, área):

self.b = b

self.h = h

self.perímetro = perímetro

self.área = área

perímetro=Rectángulo(b, h, perímetro, área)

área=b\*h

1. Diseñar un TDA Círculo, que constara de 1 atributo, radio. ¿Qué operaciones debería tener un círculo?

#Ejercicio 9

class Círculo:

def \_\_init\_\_(self, radio):

self.radio = radio

self.perímetro = perímetro

self.área = área

perímetro=Círculo(b, h, perímetro, área)

área=b\*h

1. Diseñar un TDA Vehículo. Determine que atributos tendrá y que operaciones le serían necesarias.

#Ejercicio 10

class Vehículo:

def \_\_init\_\_(self):

self.marca="Fiat"

self.tipo="Punto"

self.modelo="2011"

self.color="gris"

def auto(self):

#def \_\_auto\_\_(self):

print ("auto")

print ("auto")

1. ¿Qué diferencia hay entre un tipo de dato, una estructura de datos y un tipo de datos abstracto? Describa cada una

**Un tipo de dato informático** es un atributo de los datos que indica al ordenador (y/o al programador) sobre la clase de datos que se va a trabajar. Esto incluye imponer restricciones en los datos, como qué valores pueden tomar y qué operaciones se pueden realizar.

**Una estructura de datos** está representada por una forma determinada que tenemos de organizar los datos de un equipo informático para que podamos utilizarlos de la manera más efectiva posible. Dependiendo del tipo de aplicación o recurso que vayamos a usar requeriremos una estructura de datos independiente y distinta a las demás, dado que cada una encaja en el contexto de forma determinada y con una serie de objetivos.

Con estas estructuras tenemos la posibilidad de administrar todo tipo de datos sin ningún tipo de obstáculo, algo que en la actualidad se usa en la red para poder llevar a cabo, por ejemplo, los sistemas de indexado de contenidos. Y también juegan un papel clave en la creación de los mejores algoritmos, así como en su uso con lenguajes de programación que se benefician de ellas.

Un **tipo de dato abstracto (TDA) o tipo abstracto de datos (TAD)** es un modelo matemático compuesto por una colección de operaciones definidas sobre un conjunto de datos para el modelo.